Automatic Object Tracking in Aerial Videos via Spatial-temporal Feature Clustering

Xiaomin Tong, Yanning Zhang, Tao Yang, and Wenguang Ma

School of Computer Science, ShaanXi Provincial Key Laboratory of Speech and Image Information Processing, Northwestern Polytechnical University, Xi'an, China
xmtongnwpu@gmail.com, ynzhang@nwpu.edu.cn, yangtaonwpu@163.com

Abstract. Automatic detecting and tracking the objects from UAV videos is very important and challenging for both tactical and security applications. We present a robust object tracking system that is able to track multiple objects robustly in UAV videos. The main characteristics of the proposed system include: (1)A novel feature clustering based multiple objects tracking framework is proposed, which performs much better than the traditional foreground-blob-tracking-based methods. (2)Optical flow features are clustered both in spatial and temporal dimension to track multiple objects robustly even in the case of multiple objects cross moving. Extensive experimental results with quantitative and qualitative analysis demonstrate the robustness and effectiveness of our algorithm.

Keywords: Multiple objects tracking, optical flow, spatial-temporal trajectory clustering.

1 Introduction

With the increasing usage of UAVs for surveillance and other applications, it is of great interest to develop a fully automatic, efficient and robust object tracking system for UAV videos [1–5]. It can be widely applied in large area surveillance, search, rescue and traffic monitoring, especially for the non-cooperative targets tracking. For example, we can use it to track a car in the urban road, rescue a man in the wood or monitor some key places etc.

Many object tracking methods for UAV videos have been developed due to its various applications. Earlier typical attempt is the COCOA system [6]. It mainly contains three steps: stabilization [7–9], frame differencing, and blob tracking. However, it usually fails when the scene zooms due to the usage of Harris corner detection. Another prominent framework is proposed in [10] and [11], which performs iterative affine model estimation for image alignment, normal flow field for motion detection, graphs for representation and maintenance of a dynamic template of the moving objects. Although, this framework could achieve fast processing speed, it cannot handle the complex zooming scene. Aryo Ibrahim etc [12] construct the MODAT framework, using the SIFT feature [13] instead of Harris corner for frame matching. Unfortunately, all the above systems are under

C. Sun et al. (Eds.): IScIDE 2013, LNCS 8261, pp. 78–85, 2013.

the tracking-by-detection framework and usually fail in the complex surveillance scene. The challenges mainly come from the abrupt discontinuities in motion caused by the UAVs fast moving, low resolution noisy imagery, cluttered background, occlusion, significant change of scale, low contrast and small size of the target. All these make the detection result unstable so as to influence the tracking result. Besides, when multiple objects move crossing each other, tracking often fails due to the limitation of data association only in spacious dimension.

Recently, many object tracking algorithm basing on on-line learning theory [14, 15] are proposed. However, most of the state-of-the-art object tracking algorithms for UAV videos are usually started manually.

In this paper, we present a novel object tracking algorithm and system for real-time UAV videos. The system detects the motion trajectory by a KLT (Kanade-Lucas-Tomasi) tracker [16–18] rather than a background model. Thus, the algorithm is more robust to scale change, illumination change and other challenging cases thanks to the stability of optical flow feature. Then we confirm the object location by clustering the feature trajectories in both spatial and temporal dimension so as to track crossing objects robustly. Our system is fully automatic and does not require choosing object manually.

The rest of this paper is organized as follows. Section 2 introduces the framework of our system. Section 3 describes the details of our algorithm. Section 4 shows the experimental results and gives the discussions and Section 5 concludes the whole paper.

2 System Overview

The framework of the proposed system is shown in Fig. 1. There are mainly two modules: (1) a KLT-based feature tracking module for creation of the candidate feature point, tracking the candidate and existing feature point, and maintain or remove the feature point from the tracker list; and (2) a tracker clustering module for filtering the valid trackers, spatial trajectories clustering and temporal trajectories clustering. The individual components of the two modules are described in the following section.

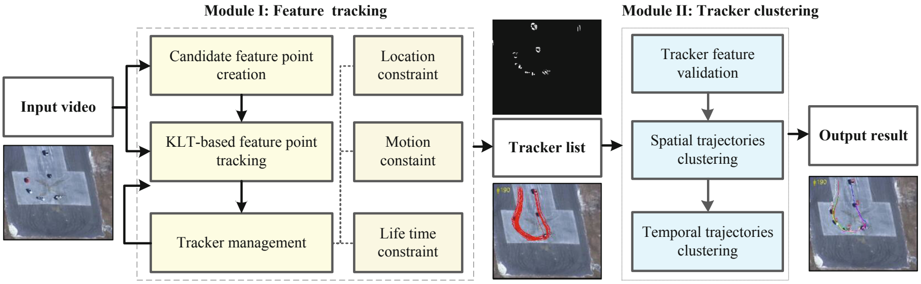
Module I: Feature tracking
Module II: Tracker clustering
Input video
Candidate feature point creation
KLT-based feature point tracking
Tracker management
Location constraint
Motion constaint
Life time constraint
Tracker list
Tracker feature validation
Spatial trajectories clustering
Temporal trajectories clustering
Output result

Fig. 1. Overview of our object tracking system

3 Spatial-temporal Clustering Based Tracking

3.1 KLT-Based Feature Tracking and Tracker Management

a) KLT based Optical flow

In order to track the object robustly in the challenging UAV videos, we adopt a KLT-based feature point tracking and clustering method instead of the background subtraction and blob tracking. For simplicity and robustness, we use the pyramidal implementation of the classical KLT tracker to get global motion vector. The core idea is to determine the local motion of window W from image I to image J and motion vector is calculated in the lowest level and propagated to higher resolution level by level. The number of pyramid levels is 5 and the patch size used is 5×5 pixels in our experiment for the trade-off between accuracy and efficiency.

b) Candidate feature creation

In an input frame, the optical flow of the good features located in the background scene accord with a transform model caused by the camera motion, while those belonging to moving object have distinguishing optical flow. Thus, we use RANSAC to get the global motion parameters and sort the candidate features that don't accord with the global motion. In our system, an affine model M is adopted to describe the global motion between two frames.

$$M = [R|T] \tag{1}$$

where R, T denote the rotation and translation parameter. Let (x_t^i, y_t^i),(x_{t+1}^i, y_{t+1}^i) represent the location of feature i at time t and $t + 1$. Then, the projection error can be calculated as follow:

$$Error = \left\| \left(x_{t+1}^i, y_{t+1}^i\right)^T - M \cdot \left(x_t^i, y_t^i, 1\right)^T \right\|_2 \tag{2}$$

If $Error$ is bigger than a given threshold, it is confirmed as a candidate feature.

By applying the pyramid KLT-based feature tracking and candidate feature point validation, an extensive feature point set with trajectories is obtained frame by frame. Meantime, we need to remove the redundant new trackers too close to the existing tracker and the wrong trackers with large motion vector.

3.2 Spatial-temporal Trajectories Clustering

As we have obtained lots of trajectories belonging to different objects, the main task is to classify them into multiple clusters. When objects are close to each other, it is difficult to obtain the objects trajectories merely via clustering the trajectories in spatial dimension. In a long term, these trajectories belonging to different object cannot be clustered into one cluster most times when objects are far away from each other, while those belonging to the same object should often have the same cluster label.

Proposition. **Two trajectories belong to the same object with high probability only if they can be clustered into the same cluster consistently in consecutive frames.**

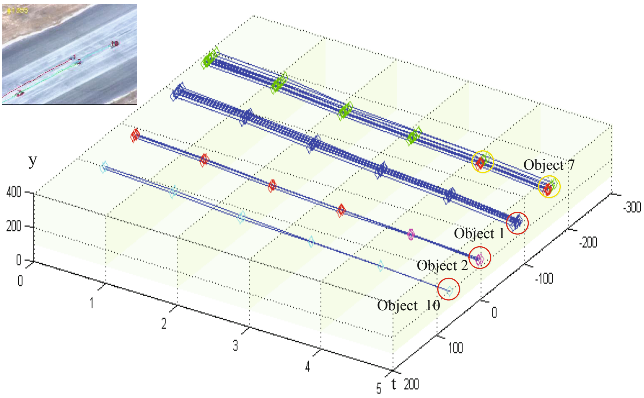

Fig. 2. Error in spatial feature clustering

For instance, in Fig. 2, there are lots of feature trajectories belonging to four objects in the red and yellow circle and the points with the same color at the same time means to belong to the same cluster. Fig. 2 shows that features belonging to object 7 are clustered to two different clusters in red and green only at time $t = 4$ and $t = 5$, while clustered into the same cluster in green at time $t = 0, ..., 3$. Thus, these features belong to the same cluster according to the Proposition. Basing on the Proposition, we first cluster the trajectories in continuous N frames separately. Valid features are filtered if its length is not less than N. Let $F(i) = (f_1(i), f_2(i), ..., f_t(i)), i = 1, 2, ..., L$ denote all the L valid feature trajectories. In order to decide the cluster number in each frame, K-means clustering is used L times with clustering number $k = 1, 2, ..., L$. With increasing of k close to the real object number, the within class variance decrease rapidly while decrease slowly when k increasing away from the real object number. Thus, we can find the turning point and obtain the cluster number K. Let $C(k, t)$ denote the center for cluster k and n_k feature points belong to cluster k.

$$C(k,t) = \frac{1}{n_k} \sum_{i=1}^{n_k} f_t(i) \tag{3}$$

where $\{f_t(i), i = 1, 2, ..., n_k\}$ refers to the n_k feature points belonging to cluster k at time t. All these feature should satisfy the following constraint.

$$|f_t(i) - C(k,t)| < \mathrm{T_o} \tag{4}$$

T_o refers to the distance threshold. For temporal clustering, we construct an association map AM with L columns and L lines. $AM(i, j)$ refers to the times $F(i)$ and $F(j)$ clustered in the same cluster in the continuous N frames. Then the maximal value in AM is N, meaning that the corresponding two features are clustered in the same cluster N times. Those small values usually refer to some wrong clustering when objects too close to each other. Thus we keep the correct association by equation (5).

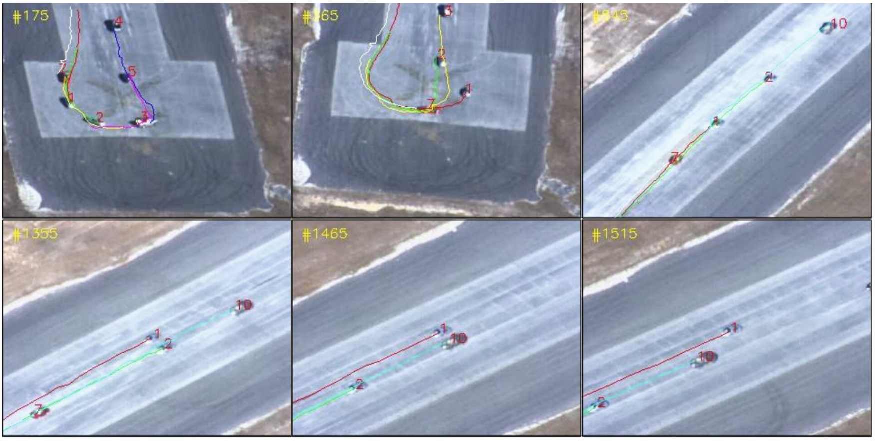

Fig. 3. Our tracking result on EgTest01 dataset

$$A(i,j)=\begin{cases}1 & if AM(i,j)>N*\alpha\\ 0 & others\end{cases} \tag{5}$$

where $\alpha \in [0,1]$ denote the association factor. We can obtain the cluster result by $A(i,j)$ and track crossing objects robustly.

4 Experimental Results

To evaluate the performance of the proposed objects tracking algorithm, extensive practical tests were undertaken on public DARPA VIVID dataset (http://vision.cse.psu.edu/data/vividEval/datasets/datasets.html). The current C++ implementation of our algorithm runs on a 3.0GHz core 2 duo machine at the rate of 7 fps for 320*240 images without any optimization. Our tracker is a fully automatical system and we can't compare our algorithm with the state-of-the-art object tracking algorithms. This is because the current trackers such as Particle Filter [15], TLD [14] are usually started manually.

Fig. 3 shows our result on EgTest01 dataset, including 1821 frames with vehicles very similar to each other. Thus, blob-based tracker is prone to fail when vehicles pass others. However, our algorithm can deal with the crossing objects tracking due to our spatial-temporal clustering. In #1355 and #1515, vehicle 1 passes vehicle 2 and vehicle 10 separately and our tracker can track the vehicles robustly under this challenging condition.

Fig. 4 gives our tracking result on EgTest02 dataset with two sets of three civilian vehicles passing by each other on a runway. As we can see, vehicle 1, vehicle 2, and vehicle 3 are tracked continuously by our algorithm even with sudden change of scale between #700 and #910. Result on RedTeam dataset is shown in Fig. 5. The challenge of tracking this vehicle mainly comes from the long shadow(#498, #1800), change of scales(between #210 and #498, #1800

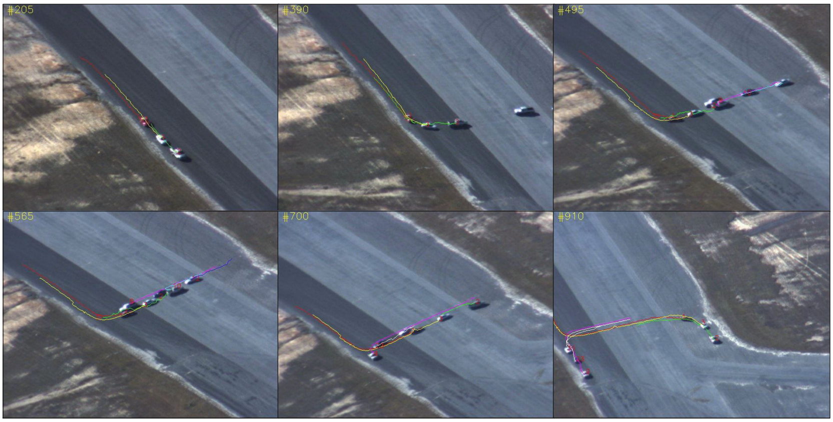

Fig. 4. Our tracking result on EgTest02 dataset

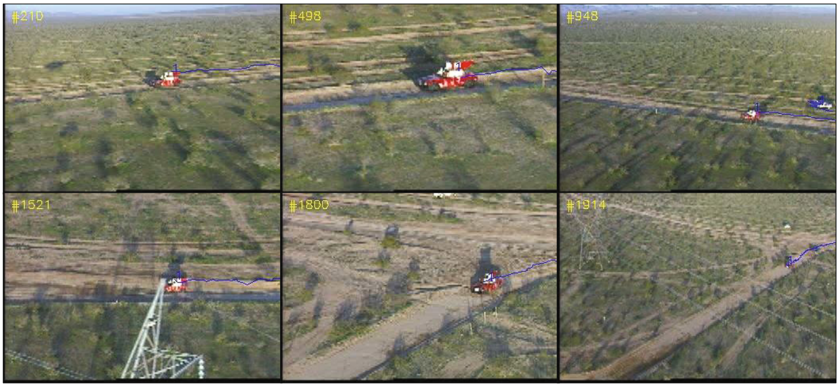

Fig. 5. Our tracking result on RedTeam dataset

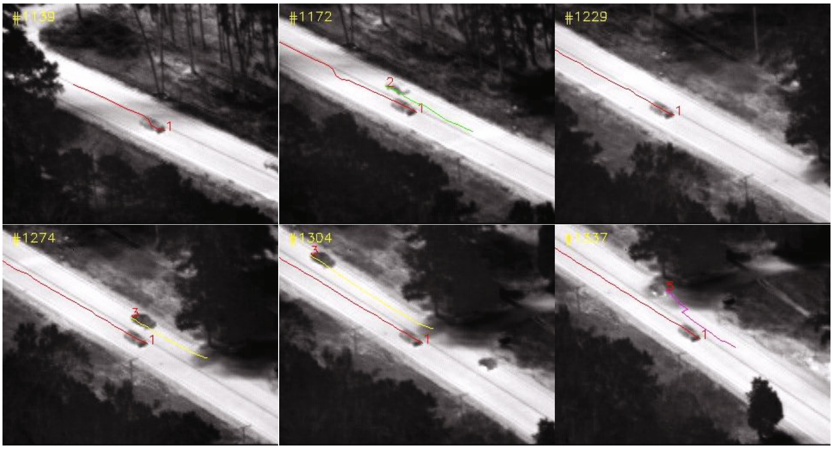

Fig. 6. Our tracking result on PkTest01 dataset

and #1914) and so on. From Fig. 5 we can see that our algorithm could perform well in these complex scenes and track objects robustly. In Fig. 6, we give the result of PkTest01 dataset which is thermal IR data of a truck. In #1172, #1274 and #1337, the truck is passed by other vehicle separately and our algorithm could track the truck continuously even other vehicle is very close to the truck.

5 Conclusion

In this paper, we present a fully automatic object tracking system for aerial video. A KLT feature tracker is adopted to estimate the feature point trajectories and spatial-temporal feature clustering is applied to cluster the feature points into different objects. Objects are tracked automatically without starting manually. Extensive experimental results on large amount of test aerial videos illustrate the robustness and efficiency of our algorithm. In the future, we will concentrate on tracking the occluded object and dealing with other challenging cases in the UAV videos.

Acknowledgment. This work is supported by the National Natural Science Foundation of China (No.61231016,No.61272288),2013 New People and New Directions Foundation of School of Computer Science in NPU(No.13GH014604), the NPU Foundation for Fundamental Research(No.JC201120,No.JC201148), and Plan of Soaring Star of Northwestern Polytechnical University(No.12GH0311).

References

1. Owen, M.,Yu, H.,McLain, T.,Beard, R.: Moving ground target tracking in urban terrain using air/ground vehicles. In: GLOBECOM Workshops (GC Wkshps), 1816–1820 (2010)
2. Radhakrishnan, G.S., Saripalli, S.: Target tracking with communication constraints: An aerial perspective. In: IEEE International Workshop on Robotic and Sensors Environments (ROSE), pp. 1–6 (2010)
3. Zhu, S., Wang, D., Low, C.B.: Ground Target Tracking Using UAV with Input Constraints. Journal of Intelligent & Robotic Systems 1-4(69), 417–429 (2013)
4. Wang, J., Zhang, Y., Lu, J., Xu, W.: A Framework for Moving Target Detection, Recognition and Tracking in UAV Videos. Affective Computing and Intelligent Interaction Advances in Intelligent and Soft Computing 137, 69–76 (2012)
5. Fu, X., Feng, H., Gao, X.: UAV Mobile Ground Target Pursuit Algorithm. Journal of Intelligent & Robotic Systems 3-4(68), 359–371 (2012)
6. Ali, S., Shah, M.: Cocoa: Tracking in Aerial Imagery. Airborne Intelligence, Surveillance, Reconnaissance (ISR) Systems and Applications III, 62090D (2006)
7. Harris, C., Stephens, M.: A Combined Corner and Edge Detector. In: Alvey Vision Conference, vol. 15, p. 50 (1988)
8. Fischler, M.A., Bolles, R.C.: Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. Communications of the ACM 24, 381–395 (1981)
9. Mann, S.: Compositing Multiple Pictures of The Same Scene. In: Proceedings of the 46th Annual IS&T Conference, vol. 2 (1993)

10. Cohen, I., Medioni, G.: Detecting and Tracking Moving Objects for Video Surveillance. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 2, pp. 319–325 (1999)
11. Medioni, G., Cohen, I., Bremond, F., Hong, S., Nevatia, R.: Event Detection and Analysis from Video Streams. IEEE Transactions on Pattern Analysis and Machine Intelligence 23, 873–889 (2001)
12. Ibrahim, A.W.N., Pang, C.W., Seet, G.L.G., Lau, W.S.M., Czajewski, W.: Moving Objects Detection and Tracking Framework for UAV-based Surveillance. In: Pacific-Rim Symposium on Image and Video Technology (PSIVT), pp. 456–461 (2010)
13. Lowe, D.G.: Object Recognition from Local Scale-Invariant Features. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV), vol. 2, pp. 1150–1157 (1999)
14. Kalal, Z., Matas, J., Mikolajczyk, K.: P-N Learning: Bootstrapping Binary Classifiers by Structural Constraints. In: Conference on Computer Vision and Pattern Recognition (2010)
15. Fan, Z., Li, M., Liu, Z.: An Improved Video Target Tracking Algorithm Based on Particle Filter and Mean-Shift. In: International Conference on Information Technology and Software Engineering, vol. 212, pp. 409–418 (2013)
16. Shi, J., Tomasi, C.: Good Features to Track. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 593–600 (1994)
17. Tomasi, C., Kanade, T.: Detection and Tracking of Point Features. Carnegie Mellon University Technical Report (1991)
18. Yang, T., Zhang, Y., Shao, D., Li, Y.: Clustering Method for Counting Passenger Getting in a Bus with Single Camera. Optical Engineering 49(037203) (2010)